

Prinsip Pemrakiraan Deret Waktu dengan Amazon Forecast



Prinsip Pemrakiraan Deret Waktu dengan Amazon Forecast: AWS Laporan Resmi

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak dapat digunakan sehubungan dengan produk atau layanan yang bukan milik Amazon, dalam bentuk apa pun yang mungkin menimbulkan kebingungan di kalangan pelanggan, atau dalam bentuk apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon adalah hak milik pemiliknya masing-masing, yang mungkin atau mungkin tidak terafiliasi dengan, berhubungan dengan, atau disponsori oleh Amazon.

Table of Contents

Abstrak dan ikhtisar	i
Ikhtisar	1
Apakah Anda Well-Architected?	2
Tentang prakiraan	3
Sistem prakiraan	3
Di mana masalah prakiraan terjadi?	3
Pertimbangan sebelum mencoba memecahkan masalah prakiraan	4
Studi kasus: Masalah prakiraan permintaan retail untuk bisnis e-commerce	6
Langkah 1: Pengumpulan dan agregat data	9
Contoh	10
Langkah 2: Persiapan data	12
Cara menangani data yang hilang	12
Contoh 1	12
Contoh 2	15
Konsep fiturisasi dan deret waktu terkait	15
Contoh 3	16
Langkah 3: Pembuatan prediktor	18
Langkah 4: Evaluasi prediktor	20
Backtesting	20
Kuantil prediksi dan metrik akurasi	22
Weighted Quantile Loss (wQL)	22
Weighted Absolute Percentage Error (WAPE)	23
Root Mean Square Error (RMSE)	23
Masalah yang terkait dengan WAPE dan RMSE	24
Langkah 5: Pembuatan dan penggunaan prakiraan untuk pengambilan keputusan	26
Prakiraan probabilistik	26
Visualisasi	27
Ringkasan alur kerja prakiraan dan API	29
Menggunakan Amazon Forecast untuk skenario umum	30
Menerapkan Prakiraan ke dalam produksi	31
Kesimpulan	32
Kontributor	33
Sumber Bacaan Lebih Lanjut	34
Lampiran A: FAQ	35

Lampiran B: Referensi	39
Riwayat dokumen	40
Pemberitahuan	41
AWS glosarium	42

Prinsip Pemrakiraan Deret Waktu dengan Amazon Forecast

Tanggal publikasi: 1 September 2021 ([Riwayat dokumen](#))

Perusahaan saat ini menggunakan segala hal, mulai dari spreadsheet yang sederhana hingga perangkat lunak perencanaan keuangan yang kompleks untuk mencoba memprakirakan hasil bisnis di masa depan secara akurat, seperti permintaan produk, kebutuhan sumber daya, dan performa keuangan. Laporan ini memperkenalkan prakiraan, serta terminologi, tantangan, dan kasus penggunaannya. Dokumen ini menggunakan studi kasus untuk mempertegas konsep pemrakiraan, langkah pemrakiraan, dan referensi tentang bagaimana [Amazon Forecast](#) dapat membantu menyelesaikan banyak tantangan praktis dalam masalah prakiraan dunia nyata.

Ikhtisar

Pemrakiraan adalah ilmu memprediksi masa depan. Dengan menggunakan data historis, bisnis dapat memahami tren, menentukan apa yang mungkin terjadi dan kapan, dan pada gilirannya, menyertakan informasi itu ke dalam rencana masa depan mereka untuk segala sesuatu mulai dari permintaan produk hingga perencanaan inventaris dan pengaturan staf.

Mengingat konsekuensi prakiraan, akurasi itu penting. Jika prakiraannya terlalu tinggi, pelanggan dapat berinvestasi berlebihan dalam produk dan staf, yang mengakibatkan investasi yang sia-sia. Jika prakiraannya terlalu rendah, pelanggan mungkin kurang berinvestasi, yang menyebabkan kekurangan bahan mentah dan inventaris, sehingga menimbulkan pengalaman pelanggan yang buruk.

Saat ini, bisnis mencoba menggunakan segala sesuatu mulai dari spreadsheet sederhana hingga perangkat lunak permintaan/perencanaan keuangan yang kompleks untuk menghasilkan prakiraan, tetapi akurasi tinggi tetap sulit didapatkan karena dua alasan:

- Pertama, prakiraan tradisional kesulitan menyertakan data historis dalam volume besar, sehingga kehilangan sinyal penting dari masa lalu yang tidak dapat ditemukan di tengah data yang salah.
- Kedua, prakiraan tradisional jarang menyertakan data terkait tetapi independen, yang dapat memberikan konteks penting (seperti harga, liburan/acara, stok habis, promosi pemasaran, dan sebagainya). Tanpa riwayat lengkap dan konteks yang lebih luas, sebagian besar prakiraan gagal memprediksi masa depan secara akurat.

[Amazon Forecast](#) adalah layanan terkelola penuh yang mengatasi masalah ini. Amazon Forecast menyediakan algoritme terbaik untuk skenario prakiraan yang ada. Layanan ini bergantung pada deep learning dan machine learning (ML) modern bila sesuai untuk memberikan prakiraan yang sangat akurat. Amazon Forecast mudah digunakan dan tidak memerlukan pengalaman machine learning. Layanan ini secara otomatis menyediakan infrastruktur yang diperlukan, memproses data, dan membangun model ML kustom/pribadi yang di-host di AWS dan siap untuk membuat prediksi. Selain itu, dengan kemajuan dalam teknik machine learning yang terus berkembang dengan cepat, Amazon Forecast menyertakan teknik ini, sehingga pelanggan terus mendapatkan peningkatan akurasi dengan sedikit atau tanpa upaya tambahan di pihak mereka.

Apakah Anda Well-Architected?

[AWS Kerangka Kerja Well-Architected](#) membantu Anda memahami pro dan kontra dari keputusan yang Anda buat saat membangun sistem di cloud. Enam pilar dari Kerangka Kerja ini memungkinkan Anda mempelajari praktik terbaik arsitektur untuk merancang dan mengoperasikan sistem yang andal, aman, efisien, hemat biaya, dan berkelanjutan. Dengan menggunakan [AWS Well-Architected Tool](#), yang tersedia tanpa biaya di [AWS Management Console](#), Anda dapat meninjau beban kerja Anda terhadap praktik terbaik ini dengan menjawab serangkaian pertanyaan untuk setiap pilar.

Di [Machine Learning Lens](#), kami fokus pada cara merancang, melakukan deployment, dan mengelola beban kerja machine learning Anda di AWS Cloud. Lensa ini merupakan tambahan untuk praktik terbaik yang dijelaskan dalam Kerangka Kerja Well-Architected.

Untuk panduan ahli dan praktik terbaik lainnya untuk arsitektur cloud Anda—referensi deployment arsitektur, diagram, dan laporan resmi—lihat [AWS Pusat Arsitektur](#).

Tentang prakiraan

Dalam dokumen ini, prakiraan berarti prediksi nilai-nilai masa depan dari deret waktu: input atau output masalah adalah karakteristik deret waktu.

Sistem prakiraan

Sistem prakiraan mencakup berbagai macam pengguna:

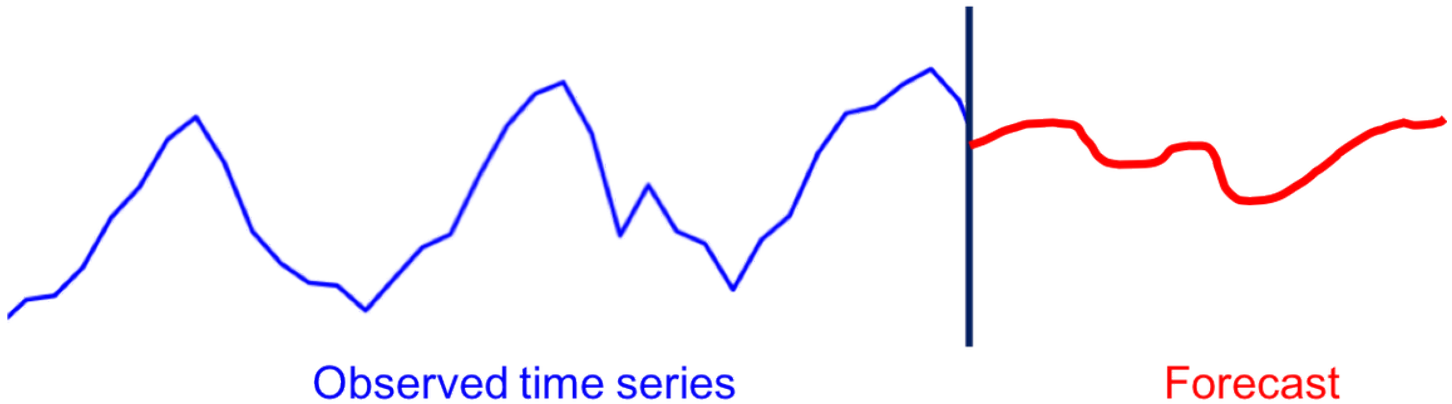
- Pengguna akhir, yang melakukan kueri prakiraan untuk produk tertentu, dan memutuskan berapa banyak unit yang akan dibeli; ini dapat berupa orang atau sistem otomatis.
- Analis bisnis/inteligensi bisnis, yang mendukung pengguna akhir, menjalankan dan mengatur pelaporan agregat.
- Ilmuwan data, yang berulang kali menganalisis pola permintaan, efek sebab akibat, dan menambahkan fitur baru untuk memberikan peningkatan inkremental pada model atau memperbaiki model prakiraan.
- Rekayasawan, yang mempersiapkan infrastruktur pengumpulan data, dan memastikan ketersediaan data input ke sistem.

Amazon Forecast meringankan pekerjaan rekayasawan perangkat lunak dan memungkinkan bisnis dengan kemampuan ilmu data terbatas untuk memanfaatkan teknologi prakiraan canggih. Untuk bisnis dengan kemampuan ilmu data, sejumlah fungsi diagnostik sudah disertakan, sehingga masalah prakiraan dapat ditangani dengan baik menggunakan Amazon Forecast.

Di mana masalah prakiraan terjadi?

Masalah prakiraan terjadi di banyak bidang yang secara alami menghasilkan data deret waktu. Ini termasuk penjualan retail, analisis medis, perencanaan kapasitas, pemantauan jaringan sensor, analisis keuangan, penambangan aktivitas sosial, dan sistem basis data. Misalnya, prakiraan memainkan peran kunci dalam mengotomatisasi dan mengoptimalkan proses operasional di sebagian besar bisnis yang memungkinkan pengambilan keputusan yang didorong data. Prakiraan untuk penawaran dan permintaan produk dapat digunakan untuk manajemen inventaris yang optimal, penjadwalan staf, dan perencanaan topologi, dan lebih umum merupakan teknologi penting untuk sebagian besar aspek pengoptimalan rantai pasokan.

Gambar berikut berisi ringkasan masalah prakiraan ketika didasarkan pada pengamatan deret waktu yang menunjukkan pola (dalam contoh ini, musiman), dan prakiraan dibuat selama jangka waktu tertentu. Sumbu horizontal mewakili waktu dari masa lalu (kiri) ke masa depan (kanan). Sumbu vertikal mewakili unit yang diukur. Dengan pertimbangan masa lalu (dalam warna biru) hingga garis hitam vertikal, mengidentifikasi masa depan (dalam warna merah) merupakan tugas prakiraan.



Ikhtisar tugas prakiraan

Pertimbangan sebelum mencoba memecahkan masalah prakiraan

Pertanyaan paling penting untuk dipahami sebelum menyelesaikan masalah prakiraan:

- Apakah Anda perlu memecahkan masalah prakiraan?
- Mengapa Anda memecahkan masalah prakiraan?

Sebagai konsekuensi dari data deret waktu yang ada di mana-mana, masalah prakiraan dapat ditemukan di mana saja. Tetapi, pertanyaan utamanya adalah apakah benar-benar ada kebutuhan untuk memecahkan masalah prakiraan, atau apakah Anda dapat menghindarinya sepenuhnya tanpa mengorbankan pengambilan keputusan yang efisien dalam bisnis. Mengajukan pertanyaan ini penting karena, secara ilmiah, prakiraan adalah salah satu masalah tersulit dalam machine learning.

Misalnya, pertimbangkan rekomendasi produk untuk peretail online. Masalah rekomendasi produk ini dapat dianggap sebagai masalah prakiraan di mana, untuk setiap pasangan unit penyimpanan stok pelanggan (SKU), Anda memperkirakan jumlah unit item tertentu yang akan dibeli oleh pelanggan tertentu ini. Formulasi masalah ini memiliki sejumlah manfaat. Salah satu manfaatnya adalah, komponen waktu diperhitungkan secara eksplisit, sehingga Anda dapat merekomendasikan produk sesuai dengan pola pembelian pelanggan.

Tetapi, masalah rekomendasi produk jarang diformulasikan sebagai masalah prakiraan, karena memecahkan masalah prakiraan seperti itu jauh lebih sulit (misalnya, kurangnya informasi di tingkat SKU pelanggan dan skala masalah) dibandingkan secara langsung memecahkan masalah rekomendasi. Oleh karena itu, ketika Anda memikirkan tentang aplikasi prakiraan, penting bagi Anda untuk mempertimbangkan penggunaan prakiraan dalam langkah selanjutnya, dan apakah masalah ini dapat diatasi menggunakan pendekatan alternatif.

[Amazon Personalize](#) dapat membantu dalam kasus ini. Amazon Personalize adalah layanan machine learning yang memudahkan developer membuat rekomendasi yang dikhususkan untuk masing-masing pelanggan menggunakan aplikasi mereka.

Setelah menentukan bahwa Anda perlu memecahkan masalah prakiraan, pertanyaan berikutnya yang harus ditanyakan adalah, mengapa Anda memecahkan masalah prakiraan? Dalam banyak lingkungan bisnis, prakiraan biasanya hanyalah sarana untuk mencapai tujuan. Misalnya, untuk prakiraan permintaan dalam konteks retail, prakiraan dapat digunakan untuk membuat keputusan manajemen inventaris. Masalah prakiraan biasanya merupakan input untuk masalah keputusan, yang pada gilirannya dapat dimodelkan sebagai masalah pengoptimalan.

Contoh masalah keputusan tersebut termasuk jumlah unit yang akan dibeli atau pendekatan terbaik untuk menangani inventaris yang ada. Masalah prakiraan bisnis lainnya termasuk memperkirakan kapasitas server atau memperkirakan permintaan untuk bahan mentah/suku cadang dalam konteks manufaktur. Prakiraan ini dapat digunakan sebagai input untuk proses lain, baik untuk masalah keputusan seperti di atas, atau untuk simulasi skenario, yang kemudian digunakan untuk perencanaan tanpa model eksplisit. Ada pengecualian terhadap aturan bahwa prakiraan bukanlah tujuannya itu sendiri. Misalnya, dalam prakiraan keuangan, prakiraan digunakan langsung untuk membangun cadangan keuangan atau untuk disampaikan kepada investor.

Untuk memahami tujuan prakiraan, pertimbangkan pertanyaan-pertanyaan berikut:

- Berapa jauh di masa depan yang harus Anda perkirakan?
- Seberapa sering Anda perlu membuat prakiraan?
- Apakah ada aspek spesifik dari prakiraan yang harus Anda pahami secara mendalam?

Studi kasus: Masalah prakiraan permintaan retail untuk bisnis e-commerce

Untuk mengilustrasikan konsep pemrakiraan dengan lebih mendetail, pertimbangkan kasus bisnis e-commerce yang menjual produk secara online. Mengoptimalkan keputusan dalam rantai pasokan (misalnya, manajemen stok) sangat penting untuk daya saing inti bisnis ini karena membantu memiliki jumlah produk yang akurat di lokasi yang sesuai untuk pemenuhan permintaan. Hal ini pada dasarnya berarti memiliki banyak pilihan yang tersedia dengan waktu pengiriman yang lebih pendek dan harga yang kompetitif, yang menghasilkan kepuasan pelanggan lebih tinggi. Input penting untuk sistem perangkat lunak rantai pasokan yakni prediksi permintaan atau prakiraan potensi penjualan setiap produk dalam katalog. Prakiraan ini memungkinkan keputusan penting di langkah selanjutnya, antara lain keputusan yang utama yakni:

- Perencanaan tingkat makro (prakiraan strategis): Untuk bisnis secara keseluruhan, apa proyeksi pertumbuhan dalam hal total penjualan/pendapatan? Di mana bisnis harus (lebih) aktif secara geografis? Bagaimana seharusnya tenaga kerja dikelola?
- Prakiraan permintaan (atau inventaris): Berapa unit dari setiap produk yang diharapkan akan dijual per lokasi?
- Aktivitas promosi (prakiraan taktis): Bagaimana seharusnya promosi dijalankan? Haruskah produk dilikuidasi?

Sisa studi kasus berfokus pada masalah kedua, yang merupakan bagian dari kelompok masalah prakiraan operasional (Januschowski & Kolassa, 2019). Dokumen ini mengikuti masalah utama: data, model (prediktor), kesimpulan (prakiraan), dan produksionisasi.

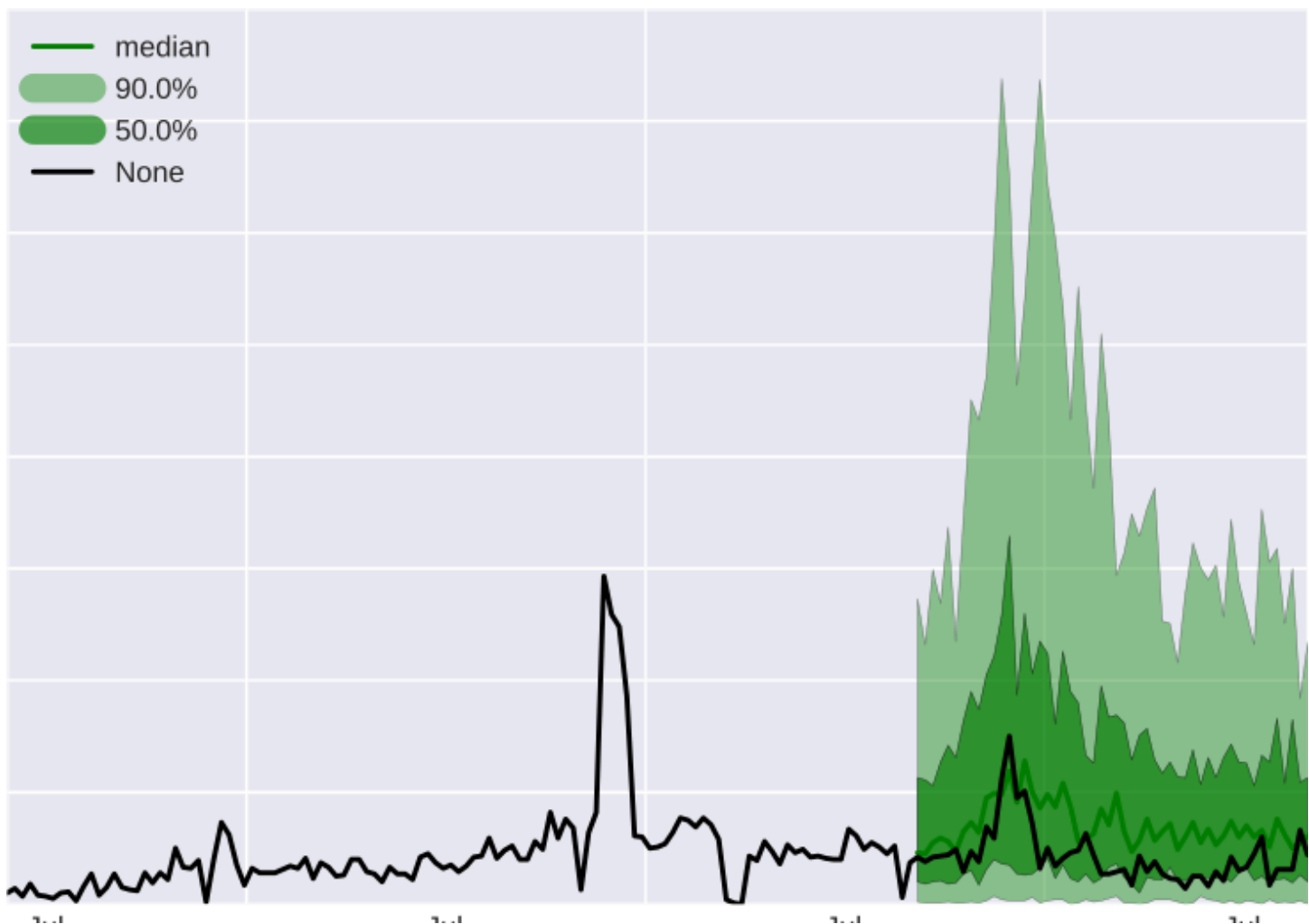
Untuk studi kasus ini, penting untuk diingat bahwa masalah prakiraan adalah sarana untuk suatu tujuan. Meskipun prakiraan sangat penting bagi bisnis, keputusan rantai pasokan di langkah selanjutnya adalah yang lebih penting. Dalam studi kasus kami, keputusan-keputusan ini diambil oleh sistem pembelian otomatis yang mengandalkan model pengoptimalan matematika dari penelitian operasi. Sistem ini mencoba meminimalkan ekspektasi biaya untuk bisnis.

Kata kuncinya yakni ekspektasi, yang berarti bahwa prakiraan seharusnya mencakup tidak hanya satu kemungkinan masa depan tetapi semua kemungkinan masa depan, dengan pembobotan yang sesuai menurut probabilitas hasil tertentu. Oleh karena itu, yang terutama memungkinkan pengambilan keputusan di langkah selanjutnya adalah distribusi penuh nilai-nilai prakiraan dan

bukannya hanya memiliki prakiraan titik. Gambar berikut menunjukkan prakiraan probabilistik (juga disebut sebagai prakiraan densitas). Perhatikan bahwa Anda dapat memperoleh prakiraan satu titik (masa depan yang paling mungkin) dengan mudah dari prakiraan probabilistik ini, tetapi beralih dari prakiraan titik ke prakiraan probabilistik akan lebih sulit.

Dengan prakiraan probabilistik, Anda dapat memperoleh statistik yang berbeda darinya dan menyesuaikan hasilnya untuk membantu mengambil keputusan yang ingin Anda ambil. Bisnis e-commerce mungkin memiliki sejumlah produk utama yang mereka hampir tidak pernah ingin kehabisan stok. Dalam hal ini, gunakan kuantil tinggi (misalnya, persentil ke-90), yang berarti ^{90%} dari waktu produk akan tersedia. Untuk produk lain, seperti produk yang penggantinya lebih mudah ditemukan (seperti pensil), menggunakan persentil yang lebih rendah mungkin lebih tepat.

Di Amazon Forecast, Anda dapat memperoleh kuantil yang berbeda dari prakiraan probabilistik dengan mudah.



Ilustrasi prakiraan probabilistik

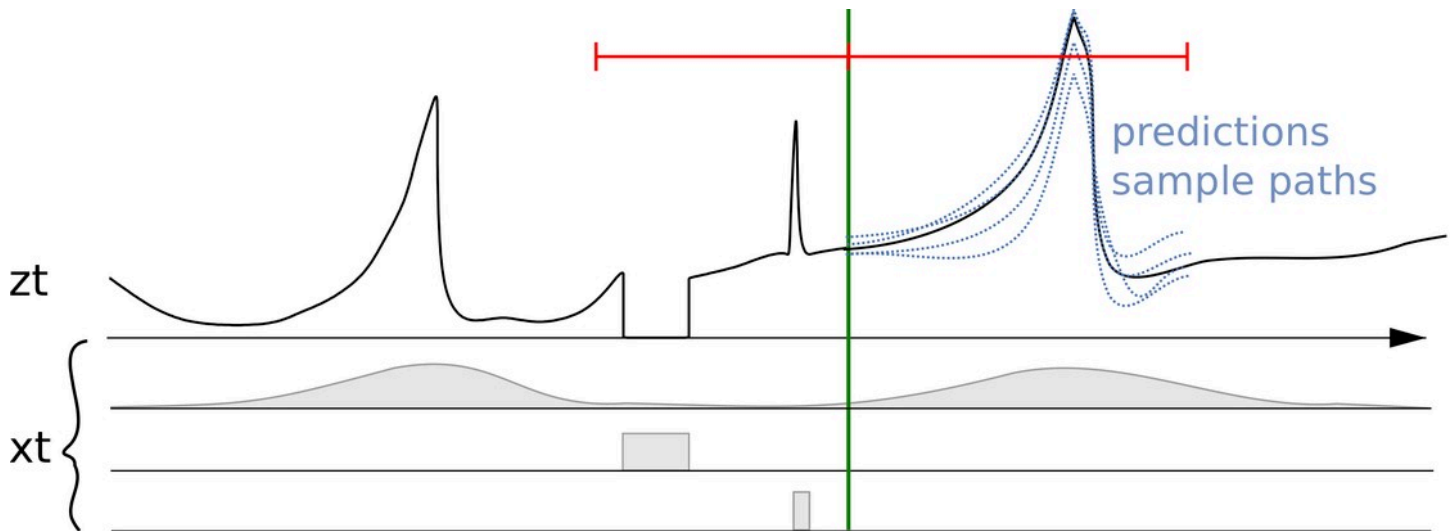
Pada gambar sebelumnya, garis hitam adalah nilai aktual; garis hijau gelap adalah median dari distribusi prakiraan; area berarsir hijau gelap adalah interval prediksi yang Anda harapkan untuk 50% dari nilai; dan area hijau muda adalah interval prediksi yang Anda harapkan untuk 90% dari nilai aktual.

Bagian berikut mencakup langkah-langkah yang terlibat untuk memecahkan masalah prakiraan untuk bisnis ini, termasuk:

- [Pengumpulan dan agregasi data \(Langkah 1\)](#)
- [Persiapan data \(Langkah 2\)](#)
- [Membuat prediktor \(Langkah 3\)](#)
- [Mengevaluasi prediktor \(Langkah 4\)](#)
- [Mengotomatiskan pembuatan prakiraan \(Langkah 5\)](#)

Langkah 1: Pengumpulan dan agregat data

Gambar berikut menunjukkan model mental untuk masalah prakiraan. Tujuannya adalah untuk memperkirakan deret waktu z_t ke masa depan, menggunakan sebanyak mungkin informasi yang relevan untuk membuat prakiraan seakurat mungkin. Oleh karena itu, langkah pertama dan paling penting adalah mengumpulkan sebanyak mungkin data yang benar.



Seri waktu z_t bersama dengan fitur terkait atau ko-variabel acak (x_t) dan beberapa prakiraan

Pada gambar sebelumnya, beberapa prakiraan ditampilkan di sebelah kanan garis vertikal. Prakiraan ini adalah contoh dari distribusi prakiraan probabilistik (atau, sebaliknya, dapat digunakan untuk mewakili prakiraan probabilistik).

Informasi utama bisnis retail untuk dicatat adalah:

- Data penjualan transaksi — Misalnya, unit penyimpanan stok (SKU), lokasi, tanda waktu, dan unit terjual.
- Data mendetail item SKU — Metadata item. Contohnya termasuk warna, departemen, ukuran, dan sebagainya.
- Data harga — Deret waktu harga setiap item dengan tanda waktu.
- Data informasi promosi — Berbagai jenis promosi, baik untuk sekumpulan item (kategori) atau masing-masing item dengan tanda waktu.
- Data informasi stok — Untuk setiap unit waktu, informasi tentang apakah SKU ada dalam stok atau dapat dibeli dibandingkan SKU tidak ada dalam stok.

- Data lokasi — Lokasi item atau penjualan pada suatu titik waktu tertentu dapat direpresentasikan sebagai tali `location_id` atau `store_id` sebagai geolokasi aktual. Geolokasi dapat berupa kode negara ditambah kode pos lima digit, atau `latitude_longitude` koordinat. Lokasi dianggap sebagai “dimensi” penjualan transaksional.

Di [Amazon Forecast](#), data historis kuantitas yang akan diperkirakan disebut Target Time Series (TTS). Untuk bisnis retail, TTS adalah data penjualan transaksional. Data historis lainnya, yang dikenal pada waktu yang persis sama dengan setiap transaksi penjualan, disebut Related Time Series (RTS). Untuk bisnis retail, RTS akan mencakup variabel harga, promosi, dan stok.

Perhatikan bahwa informasi stok penting karena masalah ini berpusat pada prakiraan permintaan dan bukan penjualan, tetapi bisnis hanya mencatat penjualan. Ketika SKU tidak dalam stok, jumlah penjualan lebih rendah dari potensi permintaan, jadi penting untuk mengetahui dan mencatat kapan peristiwa tidak dalam stok tersebut terjadi.

Set data lain yang perlu dipertimbangkan termasuk jumlah kunjungan ke halaman web, informasi mendetail tentang istilah pencarian, media sosial, dan informasi cuaca. Sering kali memiliki data yang tersedia untuk masa lalu dan untuk masa depan itu penting untuk dapat menggunakan data ini dalam model. Hal ini adalah persyaratan dari banyak model prakiraan dan backtesting (dijelaskan di bagian [Langkah 4: Evaluasi prakiraan](#)).

Untuk beberapa masalah prakiraan, frekuensi data mentah secara alami sesuai dengan masalah prakiraan. Contohnya antara lain, permintaan volume server, yang diambil sampelnya berdasarkan menit, saat Anda ingin memperkirakan frekuensi menit.

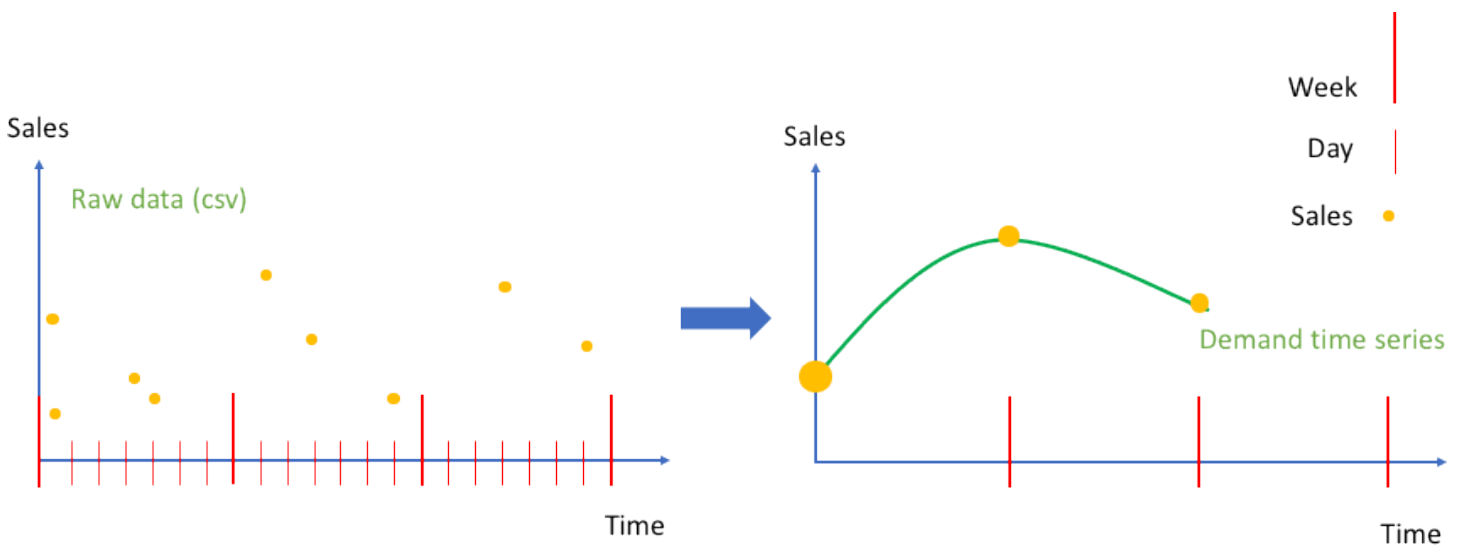
Data sering dicatat pada frekuensi yang lebih kecil, atau hanya pada tanda waktu acak dalam suatu rentang waktu, tetapi masalah prakiraan berada pada granularitas lebih besar. Hal ini umum terjadi dalam studi kasus retail, di mana data penjualan biasanya dicatat sebagai data transaksional; misalnya, format terdiri dari tanda waktu dengan granularitas sangat kecil ketika penjualan terjadi. Dalam kasus penggunaan prakiraan, granularitas rendah ini mungkin tidak diperlukan, dan mungkin lebih tepat untuk mengumpulkan data ini menjadi penjualan per jam atau harian. Di sini, tingkat agregasi sesuai dengan masalah di langkah selanjutnya; misalnya, manajemen inventaris atau perencanaan sumber daya.

Contoh

Pada gambar berikut, grafik kiri menunjukkan contoh data penjualan mentah pelanggan yang dapat dimasukkan ke Amazon Forecast sebagai file comma separated value (CSV). Dalam contoh ini, data

penjualan ditetapkan pada kisi waktu harian yang lebih kecil, dan masalahnya adalah memperkirakan permintaan mingguan pada kisi waktu yang lebih besar ke masa depan. Amazon Forecast melakukan agregasi nilai harian pada minggu tertentu dalam panggilan API `create_predictor`.

Hasilnya mengubah data mentah menjadi kumpulan deret waktu yang terbentuk dengan baik dengan frekuensi mingguan tetap. Grafik kanan menggambarkan agregasi ini pada deret waktu target menggunakan metode agregasi penjumlahan default. Metode agregasi lainnya termasuk rata-rata, maksimum, minimum, atau memilih satu titik (misalnya, yang pertama). Granularitas dan metode agregasi harus dipilih sedemikian rupa sehingga paling cocok dengan kasus penggunaan data bisnis. Dalam contoh ini, nilai agregat selaras dengan agregasi mingguan. Metode agregasi lainnya dapat diatur oleh pengguna menggunakan tombol `FeaturizationMethodParameters` dari parameter `FeaturizationConfig` di API `create_predictor`.



Agregasi data penjualan mentah sebagai peristiwa (kiri), ke deret waktu yang berdurasi sama (kanan)

Langkah 2: Persiapan data

Setelah Anda memiliki data mentah, Anda harus menangani komplikasi, seperti data yang hilang, dan pastikan Anda mempersiapkan data untuk model prakiraan yang paling tepat menangkap interpretasi yang dimaksudkan.

Cara menangani data yang hilang

Kejadian umum dalam masalah prakiraan dunia nyata adalah adanya nilai yang hilang dalam data mentah. Nilai yang hilang dalam deret waktu berarti bahwa nilai bersangkutan yang benar pada setiap titik waktu dengan frekuensi yang ditentukan tidak tersedia untuk diproses lebih lanjut. Ada beberapa alasan di balik nilai-nilai yang ditandai sebagai nilai hilang.

Nilai yang hilang dapat terjadi karena tidak ada transaksi, atau kemungkinan kesalahan pengukuran (misalnya, karena layanan yang memantau data tertentu tidak berfungsi dengan benar atau karena pengukuran tidak dapat terjadi dengan benar). Contoh utama untuk pengukuran tidak dapat terjadi dengan benar dalam studi kasus retail adalah situasi kehabisan stok dalam prakiraan permintaan, yang berarti bahwa permintaan tidak sama dengan penjualan pada hari itu.

Efek serupa dapat terjadi dalam skenario komputasi cloud ketika layanan telah mencapai batas (misalnya, instans [Amazon EC2](#) dalam [Wilayah AWS](#) tertentu semuanya sibuk). Contoh lain dari nilai yang hilang terjadi ketika produk atau layanan belum diluncurkan, atau tidak lagi diproduksi.

Nilai yang hilang juga dapat disisipkan oleh komponen pemrosesan fitur, untuk memastikan panjang deret waktu yang sama dengan padding. Jika cukup lazim, nilai yang hilang dapat berdampak signifikan pada akurasi model.

Contoh 1

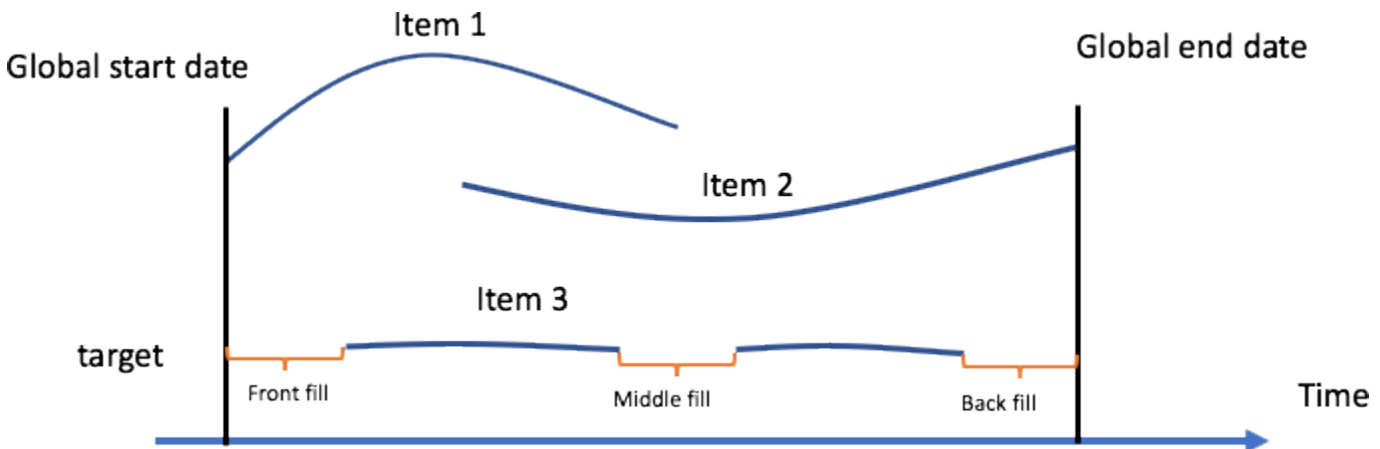
Pengisian adalah proses menambahkan nilai standar ke entri yang hilang di set data Anda. Pada gambar berikut, berbagai strategi yang berbeda untuk menangani nilai yang hilang di Amazon Forecast—pengisian depan, tengah, belakang, dan masa depan—diilustrasikan untuk item 2 dalam set data yang terdiri dari tiga item.

Amazon Forecast mendukung pengisian untuk target dan deret waktu terkait. Tanggal mulai global ditetapkan sebagai tanggal mulai paling awal dari tanggal mulai semua item dalam set data Anda. Pada contoh di bawah ini, tanggal mulai global terjadi untuk item 1. Demikian pula, tanggal akhir

global ditetapkan sebagai tanggal akhir terbaru dari deret waktu atas semua item, yang terjadi untuk item 2.

Pengisian depan mengisi setiap nilai dari awal deret waktu tertentu hingga tanggal mulai global. Pada saat penerbitan dokumen ini, Amazon Forecast tidak mengaktifkan pengisian depan apa pun, dan Amazon Forecast memungkinkan semua deret waktu dimulai pada titik waktu yang berbeda. Pengisian tengah menunjukkan nilai yang telah diisi di tengah deret waktu (misalnya, antara tanggal mulai dan tanggal akhir item), dan pengisian belakang mengisi dari tanggal terakhir deret waktu tersebut hingga tanggal akhir global.

Untuk deret waktu target, metode pengisian tengah dan belakang memiliki logika pengisian default nol. Pengisian masa depan (yang hanya berlaku untuk deret waktu terkait) mengisi nilai yang hilang antara tanggal akhir global item dan jangkauan prakiraan yang ditentukan oleh pelanggan. Nilai masa depan diperlukan untuk menggunakan set data deret waktu terkait dengan [Prophet](#) dan [DeepAR+](#), dan opsional untuk [CNN-QR](#).



Strategi penanganan nilai yang hilang di Amazon Forecast

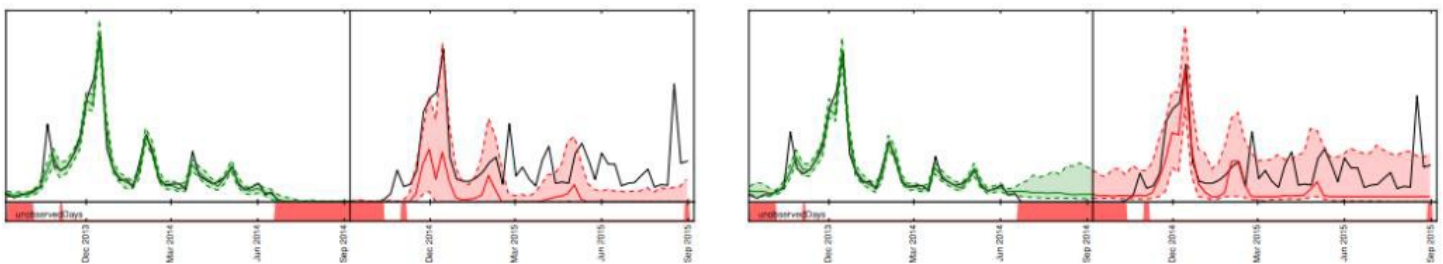
Pada gambar sebelumnya, tanggal mulai global menunjukkan tanggal mulai paling awal dari tanggal mulai semua item, dan tanggal akhir global menunjukkan tanggal akhir terbaru dari tanggal akhir semua item. Jangkauan Prakiraan adalah periode prediksi untuk nilai target yang diberikan Prakiraan.

Ini adalah skenario umum dalam studi retail yang mewakili nol penjualan untuk data transaksional untuk item yang tersedia. Nilai-nilai ini diperlakukan sebagai nol yang sebenarnya, dan digunakan dalam komponen evaluasi metrik. Amazon Forecast memungkinkan pengguna mengidentifikasi nilai yang benar-benar hilang dan menuliskannya dalam kode sebagai bukan angka (NaN) untuk diproses oleh algoritme. Laporan resmi ini selanjutnya meneliti mengapa kedua kasus ini berbeda dan kapan masing-masing kasus bermanfaat.

Dalam studi kasus retail, informasi bahwa peretail menjual nol unit item yang tersedia berbeda dari informasi bahwa nol unit barang yang tidak tersedia dijual, baik dalam periode di luar keberadaannya (misalnya, sebelum diluncurkan atau setelah dihentikan), atau dalam periode dalam keberadaannya (misalnya, sebagian kehabisan stok, atau ketika tidak ada data penjualan yang tercatat untuk rentang waktu ini). Pengisian nol default berlaku dalam kasus sebelumnya. Dalam kasus terakhir, meskipun nilai target yang bersangkutan biasanya nol, ada informasi tambahan yang disampaikan dalam nilai yang ditandai sebagai nilai hilang. Praktik terbaiknya yaitu mempertahankan informasi bahwa ada data yang hilang dan tidak membuang informasi ini. Lihat contoh berikut untuk ilustrasi mengapa menyimpan informasi itu penting.

Amazon Forecast mendukung logika pengisian tambahan nilai, rata-rata, median, minimum, dan maksimum. Untuk deret waktu terkait (misalnya, harga atau promosi), tidak ada default yang ditentukan untuk metode pengisian tengah, belakang, atau masa depan, karena logika nilai hilang yang benar bervariasi menurut jenis atribut dan kasus penggunaan. Logika pengisian yang didukung untuk deret waktu terkait termasuk nol, nilai, rata-rata, median, minimum, dan maksimum.

Untuk melakukan pengisian nilai yang hilang, tentukan jenis pengisian yang akan diterapkan saat Anda memanggil operasi [CreatePredictor](#). Logika pengisian ditentukan dalam objek [FeaturizationMethod](#). Misalnya, untuk menuliskan dalam kode nilai yang tidak mewakili nol penjualan produk yang tidak tersedia dalam deret waktu target, tandai nilai sebagai benar-benar hilang dengan menetapkan jenis pengisian yang sama dengan NaN. Tidak seperti pengisian nol, nilai yang dikodekan NaN diperlakukan sebagai benar-benar hilang, dan tidak digunakan dalam komponen evaluasi metrik



Efek pengisian 0 vs pengisian dengan NaN pada prakiraan untuk item yang sama

Pada gambar sebelumnya, pada grafik kiri, nilai di sebelah kiri garis hitam vertikal diisi dengan 0, yang menghasilkan prakiraan underbias (di sebelah kanan garis hitam vertikal). Pada grafik kanan, nilai-nilai ini ditandai sebagai NaN, yang menghasilkan prakiraan yang tepat.

Contoh 2

Gambar sebelumnya menggambarkan pentingnya penanganan nilai yang hilang dengan benar untuk model linear state space, seperti [ARIMA atau ETS](#). Gambar ini berisi prakiraan permintaan untuk item yang sebagian kehabisan stok. Wilayah pelatihan ditampilkan di grafik kiri berwarna hijau, rentang prediksi di panel kanan berwarna merah, dan target sebenarnya dalam warna hitam. Prakiraan median, p10, dan p90 masing-masing ditampilkan di garis merah dan wilayah berarsir. Bagian bawah menunjukkan item yang kehabisan stok (80% dari data) yang ditandai dengan warna merah. Di grafik kiri, area kehabisan stok diabaikan dan diisi dengan 0.

Ini menghasilkan model prakiraan dengan asumsi bahwa ada banyak angka nol untuk diprediksi, oleh karena itu prakiraannya terlalu rendah. Pada grafik kanan, area kehabisan stok diperlakukan sebagai pengamatan yang benar-benar hilang, dan permintaan menjadi tidak pasti di wilayah kehabisan stok. Dengan nilai yang hilang untuk item kehabisan stok yang ditandai dengan benar sebagai NaN, Anda tidak melihat underbias dalam rentang prediksi pada grafik ini. Amazon Forecast mengisi kesenjangan ini dalam data, sehingga memudahkan Anda menangani data yang hilang dengan benar, tanpa harus secara eksplisit memodifikasi semua data input mereka.

Konsep fiturisasi dan deret waktu terkait

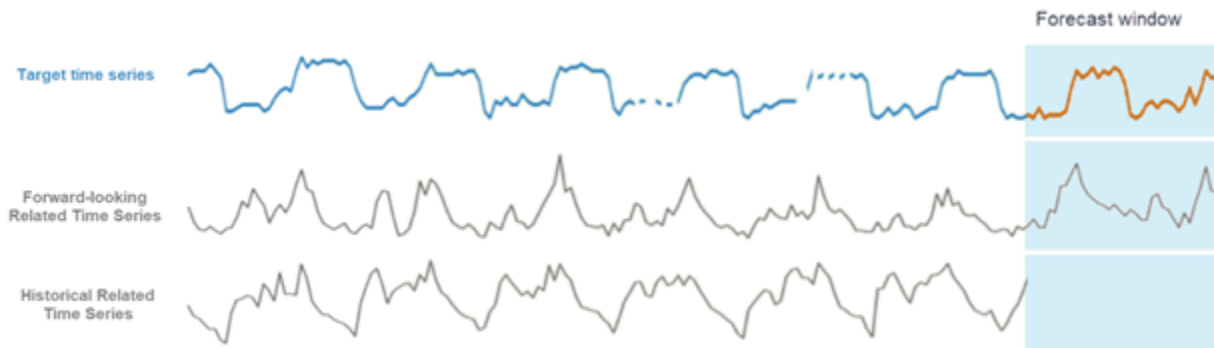
Amazon Forecast memungkinkan pengguna memasukkan data terkait untuk membantu meningkatkan akurasi model prakiraan tertentu yang didukung. Ada dua jenis wujud data ini: deret waktu terkait atau metadata item statis.

Note

Metadata dan Data terkait disebut sebagai fitur dalam machine learning, dan kovariat dalam statistik.

Deret waktu terkait adalah deret waktu yang memiliki korelasi dengan nilai target, dan seharusnya memberikan kekuatan statistik untuk memperkirakan nilai target karena memberikan penjelasan dalam istilah intuitif (lihat [Amazon Forecast: memprediksi deret waktu dalam skala besar](#) sebagai contoh). Berbeda dengan deret waktu target, deret waktu terkait adalah nilai yang diketahui di masa lalu yang dapat memengaruhi deret waktu target, dan mungkin memiliki nilai yang diketahui di masa depan.

Di Amazon Forecast, Anda dapat menambahkan dua jenis deret waktu terkait: deret waktu historis dan deret waktu berwawasan ke depan. Deret waktu terkait historis berisi titik data hingga jangkauan prakiraan, dan tidak berisi titik data apa pun dalam jangkauan prakiraan di masa depan. Deret waktu terkait dengan wawasan ke depan berisi titik data hingga dan dalam jangkauan prakiraan.



Pendekatan berbeda sehubungan penggunaan deret waktu terkait dengan Amazon Forecast

Contoh 3

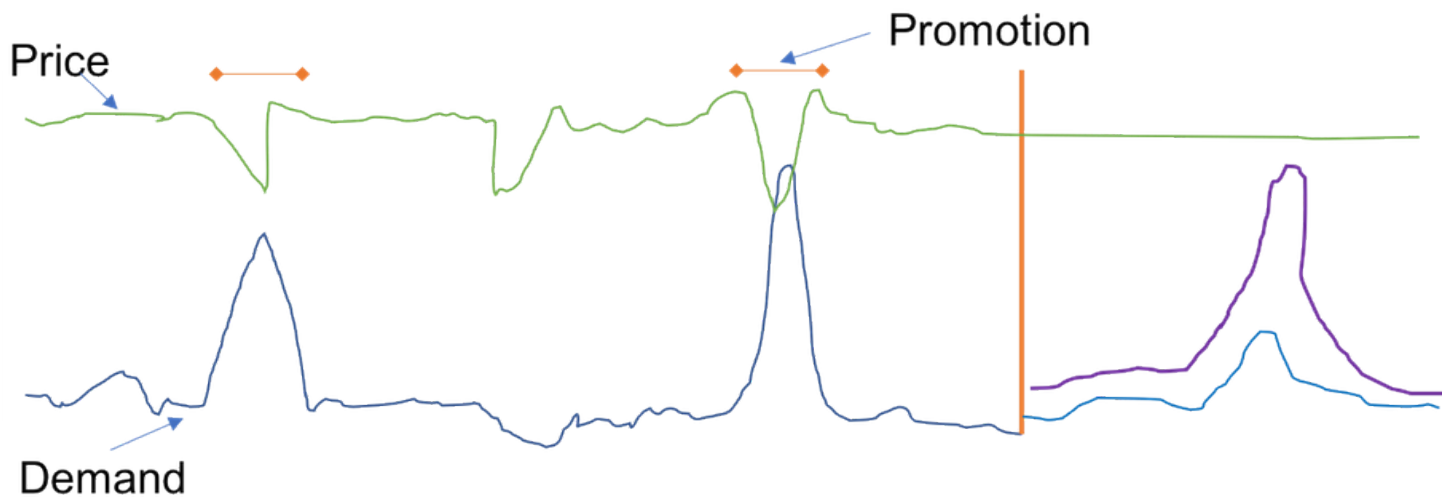
Gambar berikut menunjukkan contoh bagaimana deret waktu terkait dapat digunakan untuk memprediksi permintaan akan sebuah buku populer di masa depan. Garis biru mewakili permintaan dalam deret waktu target. Harga ditampilkan sebagai garis hijau. Garis vertikal mewakili tanggal mulai prakiraan, dan prakiraan dalam dua kuantil ditampilkan di sebelah kanan garis vertikal.

Contoh ini menggunakan deret waktu terkait dengan wawasan ke depan yang selaras dengan deret waktu target pada granularitas prakiraan, dan yang diketahui pada setiap waktu (atau hampir setiap waktu) di masa depan dalam rentang tanggal mulai prakiraan hingga tanggal mulai prakiraan yang ditingkatkan dengan jangkauan prakiraan (tanggal akhir prakiraan).

Gambar berikut juga menunjukkan bahwa harga adalah fitur yang cocok untuk digunakan, karena Anda dapat melihat korelasi antara penurunan harga dengan peningkatan penjualan produk. Deret waktu terkait dapat diberikan ke Amazon Forecast melalui file CSV terpisah, yang berisi SKU item, tanda waktu, dan nilai deret waktu terkait (dalam hal ini, harga).

Amazon Forecast mendukung metode agregasi, seperti rata-rata dan penjumlahan untuk deret waktu target, tetapi tidak untuk deret waktu terkait. Misalnya, tidak masuk akal untuk menjumlahkan harga harian dengan harga mingguan, demikian juga untuk promosi harian.

Amazon Forecast dapat secara otomatis menyertakan informasi [cuaca](#) dan [hari libur](#) ke dalam model dengan menyertakan set data fitur bawaan (lihat [SupplementaryFeature](#)). Informasi cuaca dan hari libur dapat secara signifikan memengaruhi permintaan retail.



Penjualan item tertentu (dengan warna biru, di sebelah kiri garis merah vertikal)

Metadata item, juga dikenal sebagai variabel kategoris, adalah fitur bermanfaat lainnya yang dapat dimasukkan ke Amazon Forecast (lihat [Amazon Forecast: memprediksi deret waktu dalam skala besar](#), sebagai contoh). Perbedaan utama antara variabel kategoris dan deret waktu terkait adalah bahwa variabel kategoris bersifat statis - variabel kategoris tidak berubah seiring waktu. Contoh retail umum antara lain warna item, kategori buku, dan indikator biner apakah TV adalah TV pintar atau tidak. Informasi ini dapat diambil oleh algoritme deep learning dalam mempelajari kesamaan antara unit penyimpanan stok (SKU), dengan asumsi bahwa SKU serupa memiliki penjualan yang serupa. Karena metadata ini tidak memiliki ketergantungan waktu, setiap baris dalam file CSV metadata item hanya terdiri dari SKU item dan label atau deskripsi kategori yang bersangkutan.

Langkah 3: Pembuatan prediktor

Prediktor dapat dibuat dalam dua cara: menjalankan [AutoML](#) atau secara manual memilih salah satu dari enam algoritme bawaan Amazon Forecast. Saat menjalankan AutoML, pada saat penulisan dokumen ini, Amazon Forecast secara otomatis menguji enam algoritme bawaan dan memilih algoritme yang memiliki kesalahan kuantil rata-rata terendah dalam kuantil ke-10, ke-50 (median), dan ke-90.

Amazon Forecast menawarkan empat model lokal:

- Autoregressive Integrated Moving Average ([ARIMA](#))
- Exponential Smoothing ([ETS](#))
- Non-Parametric Time Series ([NPTS](#))
- [Prophet](#)

Model lokal adalah metode prakiraan yang sesuai dengan satu model untuk setiap deret waktu individu (atau kombinasi item/dimensi tertentu), kemudian menggunakan model tersebut untuk mengekstrapolasi deret waktu ke masa depan.

ARIMA dan ETS adalah versi model lokal populer yang dapat diskalakan dari paket prakiraan R. NPTS, metode lokal yang dikembangkan di Amazon, memiliki perbedaan utama jika dibandingkan dengan model lokal lainnya. Berbeda dengan prediktor musiman sederhana, yang memberikan prakiraan titik dengan mengulangi nilai terakhir atau nilai pada musiman yang sesuai, NPTS menghasilkan prakiraan probabilistik. NPTS menggunakan indeks waktu tetap, di mana indeks sebelumnya ($T - 1$) atau musim lalu ($T - \tau$) adalah prediksi untuk langkah waktu T . Algoritme secara acak mengambil sampel indeks waktu (t) di set $\{0, \dots, T - 1\}$ untuk menghasilkan sampel untuk langkah waktu T saat ini. NPTS sangat efektif untuk deret waktu intermiten (kadang-kadang juga disebut jarang) dengan banyak nol. Prakiraan juga mencakup implementasi Python Prophet, model deret waktu struktural Bayesian.

Amazon Forecast menawarkan dua algoritme deep learning global:

- [DeepAR+](#)
- [CNN-QR](#)

Model global melatih satu model di seluruh koleksi deret waktu dalam set data. Hal ini sangat berguna ketika ada deret waktu yang serupa di satu set unit lintas sektional. Misalnya, pengelompokan deret waktu permintaan untuk permintaan halaman web, beban server, dan produk yang berbeda.

Secara umum, seiring bertambahnya jumlah deret waktu, efikasi CNN-QR dan DeepAR+ meningkat. Hal ini tidak selalu terjadi untuk model lokal. Model deep learning juga dapat digunakan untuk menghasilkan prakiraan untuk SKU baru dengan sedikit atau tanpa data penjualan historis. Hal ini dikenal sebagai [prakiraan Cold Start](#).

	Neural Networks		Flexible Local Algorithms	Baseline Algorithms		
	CNN-QR	DeepAR+	Prophet	NPTS	ARIMA	ETS
Computationally intensive training process	High	High	Medium	Low	Low	Low
Accepts historical related time series*	✔	✘	✘	✘	✘	✘
Accepts forward-looking related time series*	✔	✔	✔	✘	✘	✘
Accepts item metadata (product color, brand, etc)	✔	✔	✘	✘	✘	✘
Suitable for sparse datasets	✔	✔	✘	✔	✘	✘
Performs Hyperparameter Optimization (HPO)	✔	✔	✘	✘	✘	✘
Allows overriding default hyperparameter values	✔	✔	✘	✔	✘	✘
Suitable for What-if analysis	✔	✔	✔	✘	✘	✘
Suitable for Cold Start scenarios (forecasting with little to no historical data)	✔	✔	✘	✘	✘	✘

Bandingkan algoritme yang tersedia di Amazon Forecast

Untuk informasi lebih lanjut tentang deret waktu terkait, lihat [Deret Waktu Terkait](#).

Langkah 4: Evaluasi prediktor

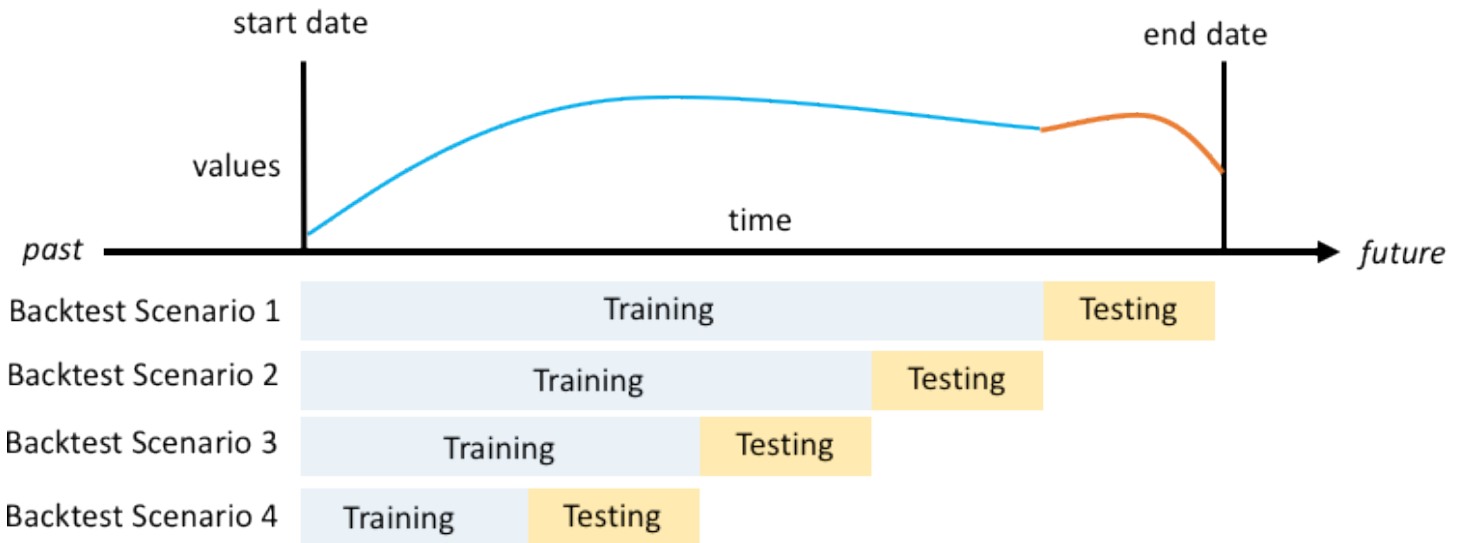
Alur kerja khas dalam machine learning terdiri dari pelatihan serangkaian model atau kombinasi model pada set pelatihan dan penilaian keakuratannya pada set data penahanan. Bagian ini membahas cara membagi data historis, dan metrik mana yang digunakan untuk mengevaluasi model dalam prakiraan deret waktu. Untuk prakiraan, teknik backtesting adalah alat utama untuk menilai akurasi prakiraan.

Backtesting

Kerangka kerja backtesting dan evaluasi yang tepat adalah salah satu faktor terpenting dalam meraih kesuksesan aplikasi machine learning. Anda dapat mengandalkan backtest yang sukses dengan model Anda untuk mendapatkan keyakinan atas kekuatan prediktif masa depan model. Selain itu, Anda dapat menyesuaikan model melalui hyper-parameter optimization (HPO), mempelajari kombinasi model, dan mengaktifkan meta-learning dan AutoML.

Waktu karakteristik prakiraan deret waktu membuatnya berbeda, dalam hal metodologi backtesting dan evaluasi, dari bidang lain machine learning terapan. Biasanya dalam tugas ML, untuk menilai kesalahan prediktif dalam backtest, Anda membagi set data berdasarkan item. Misalnya, untuk validasi silang dalam tugas terkait gambar, Anda melakukan pelatihan di beberapa persentase gambar, kemudian menggunakan bagian lain untuk pengujian dan validasi. Dalam prakiraan, Anda perlu membagi terutama berdasarkan waktu (dan pada tingkat yang lebih rendah berdasarkan item) untuk memastikan bahwa Anda tidak membocorkan informasi dari set pelatihan ke set pengujian atau validasi, dan bahwa Anda mensimulasikan kasus produksi semirip mungkin.

Pembagian berdasarkan waktu harus dilakukan dengan hati-hati karena Anda tidak ingin memilih satu titik waktu, tetapi beberapa titik waktu. Jika tidak, akurasi terlalu bergantung pada tanggal mulai prakiraan, seperti yang ditetapkan oleh titik pembagian. Evaluasi prakiraan bergulir, di mana Anda melakukan serangkaian pembagian selama beberapa titik waktu dan hasil rata-rata yang dikeluarkan memberikan hasil backtest yang lebih kuat dan dapat diandalkan. Gambar berikut menggambarkan empat pembagian backtest yang berbeda.



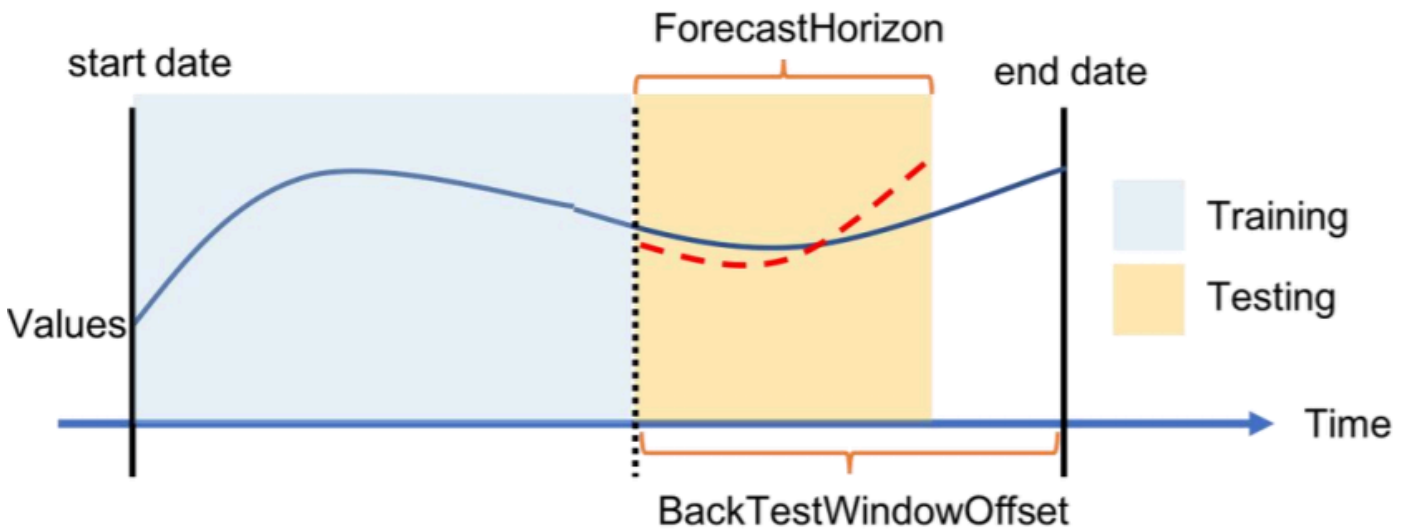
Ilustrasi empat skenario backtesting yang berbeda dengan peningkatan ukuran set pelatihan, tetapi ukuran pengujian konstan

Pada gambar sebelumnya, semua skenario backtesting memiliki data yang tersedia selama seluruh durasi untuk dapat mengevaluasi nilai yang diperkirakan terhadap aktual.

Alasan di balik beberapa jendela backtest diperlukan adalah bahwa sebagian besar deret waktu dalam dunia nyata biasanya non-stasioner. Bisnis e-commerce dalam studi kasus berkantor pusat di Amerika Utara dan sebagian besar permintaan produknya didorong oleh puncak Q4, dengan puncak tertentu pada waktu sekitar Thanksgiving dan sebelum Natal. Dalam musim belanja Q4, variabilitas deret waktu lebih tinggi daripada waktu lain sepanjang tahun ini. Dengan memiliki beberapa jendela backtest, Anda dapat mengevaluasi model prakiraan dalam pengaturan yang lebih seimbang.

Untuk setiap skenario backtest, gambar berikut menunjukkan unsur-unsur dasar dalam terminologi Amazon Forecast. Amazon Forecast secara otomatis membagi data menjadi set data pelatihan dan pengujian. Amazon Forecast memutuskan cara membagi data input dengan menggunakan `BackTestWindowOffset` parameter yang ditetapkan sebagai parameter dalam `create_predictor` API atau menggunakan nilai defaultnya `ForecastHorizon`.

Pada gambar berikut, Anda melihat kasus sebelumnya yang lebih umum ketika parameter `BackTestWindowOffset` dan `ForecastHorizon` tidak sama. Parameter `BackTestWindowOffset` menetapkan tanggal mulai prakiraan virtual, yang ditampilkan sebagai garis vertikal putus-putus pada gambar berikut. Ini dapat digunakan untuk menjawab pertanyaan hipotetis berikut: Jika model diterapkan pada hari ini, apakah prakiraannya? `ForecastHorizon` menetapkan jumlah langkah waktu dari tanggal mulai prakiraan virtual ke prediksi.



Ilustrasi skenario backtest tunggal dan konfigurasinya di Amazon Forecast

Amazon Forecast dapat mengekspor nilai prakiraan dan metrik akurasi yang dihasilkan selama backtesting. Data yang diekspor dapat digunakan untuk mengevaluasi item tertentu pada kuantil dan titik waktu tertentu.

Kuantil prediksi dan metrik akurasi

Kuantil prediksi dapat memberikan batas atas dan bawah untuk prakiraan. Misalnya, menggunakan jenis prakiraan 0,1 (P10), 0,5 (P50), dan 0,9 (P90) memberikan rentang nilai yang disebut sebagai interval kepercayaan 80% di sekitar prakiraan P50. Dengan menghasilkan prediksi pada P10, P50, dan P90, Anda dapat mengharapkan nilai sebenarnya ada di antara batas tersebut 80% dari waktu.

Laporan resmi ini lebih lanjut membahas kuantil pada [Langkah 5](#).

Amazon Forecast menggunakan metrik akurasi Weighted Quantile Loss (wQL), Root Mean Square Error (RMSE), dan Weighted Absolute Percentage Error (WAPE) untuk mengevaluasi prediktor selama backtesting.

Weighted Quantile Loss (wQL)

Metrik kesalahan Weighted Quantile Loss (wQL) mengukur akurasi prakiraan model pada kuantil yang ditentukan. Metrik ini sangat berguna ketika ada biaya yang berbeda untuk prediksi yang kurang dan prediksi berlebih. Mengatur bobot (τ) fungsi wQL secara otomatis menyertakan hukuman yang berbeda untuk prediksi yang kurang dan prediksi berlebih.

$$wQL[\tau] = 2 \frac{\sum_{i,t} [\tau \max(y_{i,t} - q_{i,t}^{(\tau)}, 0) + (1 - \tau) \max(q_{i,t}^{(\tau)} - y_{i,t}, 0)]}{\sum_{i,t} |y_{i,t}|}$$

fungsi wQL

Di mana:

- τ — Kuantil dalam set $\{0,01, 0,02, \dots, 0,99\}$
- $q_{i,t}(\tau)$ — Kuantil τ yang diprediksi model.
- $y_{i,t}$ — Nilai yang teramati di titik (i,t)

Weighted Absolute Percentage Error (WAPE)

Weighted Absolute Percentage Error (WAPE) adalah metrik yang umum digunakan untuk mengukur akurasi model. Metrik ini mengukur deviasi keseluruhan nilai yang diperkirakan dari nilai yang diamati.

$$WAPE = \frac{\sum_{i,t} |y_{i,t} - \hat{y}_{i,t}|}{\sum_{i,t} |y_{i,t}|}$$

WAPE

Di mana:

- $y_{i,t}$ - Nilai yang teramati di titik (i,t)
- $\hat{y}_{i,t}$ - Nilai yang diprediksi di titik (i,t)

Prakiraan menggunakan prakiraan rata-rata sebagai nilai yang diprediksi, $\hat{y}_{i,t}$.

Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{nT} \sum_{i,t} (\hat{y}_{i,t} - y_{i,t})^2}$$

Root Mean Square Error (RMSE) adalah metrik yang umum digunakan untuk mengukur akurasi model. Seperti WAPE, metrik ini mengukur deviasi keseluruhan perkiraan dari nilai yang diamati.

Di mana:

- $y_{i,t}$ - Nilai yang teramati di titik (i,t)
- $\hat{y}_{i,t}$ - Nilai yang diprediksi di titik (i,t)
- nT - Jumlah titik data dalam set pengujian

Prakiraan menggunakan prakiraan rata-rata sebagai nilai yang diprediksi, $\hat{y}_{i,t}$. Saat menghitung metrik prediktor, nT adalah jumlah titik data dalam jendela backtest.

Masalah yang terkait dengan WAPE dan RMSE

Dalam kebanyakan kasus, prakiraan titik yang dapat dihasilkan secara internal atau dari alat prakiraan lainnya harus sesuai dengan prakiraan rata-rata atau kuantil p50. Untuk WAPE dan RMSE, Amazon Forecast menggunakan prakiraan rata-rata untuk mewakili nilai yang diprediksi (\hat{y}).

Untuk $\tau = 0,5$ dalam persamaan $wQL[\tau]$, kedua bobot sama, dan $wQL[0,5]$ mengurangi hingga Weighted Absolute Percentage Error (WAPE) yang umum digunakan untuk prakiraan titik:

$$wQL[0.5] = 2 \frac{\sum_{i,t} 0.5 [\max(y_{i,t} - q_{i,t}^{(0.5)}, 0) + \max(q_{i,t}^{(0.5)} - y_{i,t}, 0)]}{\sum_{i,t} |y_{i,t}|} = \frac{\sum_{i,t} |y_{i,t} - q_{i,t}^{(0.5)}|}{\sum_{i,t} |y_{i,t}|}$$

di mana $\hat{y} = q(0,5)$ adalah prakiraan komputasi. Faktor penskalaan 2 digunakan dalam rumus wQL untuk membatalkan faktor 0,5 guna mendapatkan ekspresi $WAPE[\text{median}]$ yang tepat.

Perhatikan bahwa definisi WAPE di atas berbeda dari interpretasi umum untuk Mean Absolute Percentage Error ([MAPE](#)). Perbedaannya ada pada penyebutnya. Cara WAPE didefinisikan di atas menghindari masalah pembagian dengan angka 0, masalah yang sering terjadi dalam skenario dunia nyata seperti bisnis e-commerce dalam studi kasus, yang akan sering kali menjual 0 unit SKU tertentu pada hari tertentu.

Tidak seperti metrik weighted quantile loss untuk τ tidak sama dengan 0,5, bias yang ada pada setiap kuantil tidak dapat ditangkap dengan perhitungan seperti WAPE, di mana bobotnya sama. Kelemahan lain dari WAPE termasuk bahwa WAPE tidak simetris, melebih-lebihkan kesalahan persentase untuk jumlah kecil, dan hanya metrik untuk titik.

RMSE adalah kuadrat dari variabel kesalahan dalam WAPE dan merupakan metrik kesalahan umum dalam aplikasi ML lainnya. Metrik RMSE mendukung sebuah model, di mana kesalahan individu memiliki besar yang konsisten, karena variasi kesalahan yang besar akan meningkatkan RMSE secara berlebihan. Karena kesalahan yang dikuadratkan, beberapa nilai yang diprediksi dengan buruk dalam prakiraan yang baik dapat meningkatkan RMSE. Selain itu, karena variabel kuadrat, variabel kesalahan yang lebih kecil memiliki bobot yang lebih kecil di RMSE daripada di WAPE.

Metrik akurasi memungkinkan penilaian kuantitatif prakiraan. Khususnya untuk perbandingan skala besar (apakah metode A lebih baik daripada metode B secara keseluruhan), hal ini sangat penting. Tetapi, melengkapi ini dengan visual untuk SKU individu sering kali penting.

Langkah 5: Pembuatan dan penggunaan prakiraan untuk pengambilan keputusan

Setelah Anda memiliki model yang memenuhi ambang batas akurasi yang diperlukan untuk kasus penggunaan spesifik Anda (sebagaimana ditentukan melalui backtesting), langkah terakhir melibatkan deployment model dan pembuatan prakiraan. Untuk melakukan deployment model di Amazon Forecast, Anda harus menjalankan API `Create_Forecast`. Tindakan ini meng-host model yang dibuat dengan melatih seluruh set data historis (tidak seperti `Create_Predictor`, yang membagi data menjadi set pengujian dan set pelatihan). Prediksi model yang dihasilkan di jangkauan prakiraan kemudian dapat dikonsumsi dalam dua cara:

- Anda dapat melakukan kueri prakiraan untuk item tertentu (dengan menentukan item atau kombinasi item/dimensi) menggunakan API `Query_Forecast` dari [AWS CLI](#) atau secara langsung melalui [AWS Management Console](#).
- Anda dapat membuat prakiraan untuk semua kombinasi item dan dimensi di semua kuantil menggunakan API `Create_Forecast_Export_Job`. API ini menghasilkan file CSV yang disimpan dengan aman di lokasi [Amazon Simple Storage Service](#) (Amazon S3) pilihan Anda. Anda kemudian dapat menggunakan data dari file CSV dan memasukkannya ke sistem selanjutnya yang digunakan untuk pengambilan keputusan. Misalnya, sistem rantai pasokan yang ada dapat menyerap output dari Amazon Forecast secara langsung untuk membantu memberikan informasi dalam pengambilan keputusan seputar pembuatan SKU tertentu.

Prakiraan probabilistik

Amazon Forecast dapat membuat prakiraan pada kuantil yang berbeda, yang sangat berguna ketika biaya kurang prediksi dan prediksi berlebih berbeda. Mirip dengan tahap pelatihan prediktor, prakiraan probabilistik dapat dibuat untuk kuantil antara p1 dan p99.

Secara default, Amazon Forecast membuat prakiraan pada kuantil yang sama yang digunakan selama pelatihan prediktor. Jika kuantil tidak ditentukan selama pelatihan prediktor, prakiraan akan dibuat pada p10, p50, dan p90 secara default.

Untuk prakiraan p10, nilai sebenarnya diharapkan lebih rendah dari nilai yang diprediksi 10% dari waktu, dan metrik $wQL[0, 1]$ dapat digunakan untuk menilai keakuratannya. Ini berarti prakiraan P10 berada di bawah prakiraan 90% dari waktu, dan jika digunakan untuk inventaris stok, 90% dari waktu

item akan terjual habis. Prakiraan P10 bisa berguna ketika tidak ada banyak ruang penyimpanan, atau biaya modal yang diinvestasikan tinggi.

Note

Definisi formal dari prakiraan kuantil adalah $Pr(\text{nilai aktual} \leq \text{prakiraan pada kuantil } q) = q$. Secara teknis kuantil adalah persentil/100. Ahli statistik cenderung mengatakan "tingkat kuantil P90", karena itu lebih mudah dikatakan daripada "kuantil 0,9". Misalnya, prakiraan tingkat kuantil P90 berarti nilai aktual dapat diperkirakan kurang dari prakiraan 90% dari waktu. Khususnya jika pada waktu= t_1 dan tingkat quantile= 0,9, nilai prediksi = 30, itu berarti nilai aktual pada waktu= t_1 , jika Anda memiliki 1.000 simulasi, diperkirakan kurang dari 30 untuk 900 simulasi, dan untuk 100 simulasi, nilai aktual diperkirakan lebih dari 30.

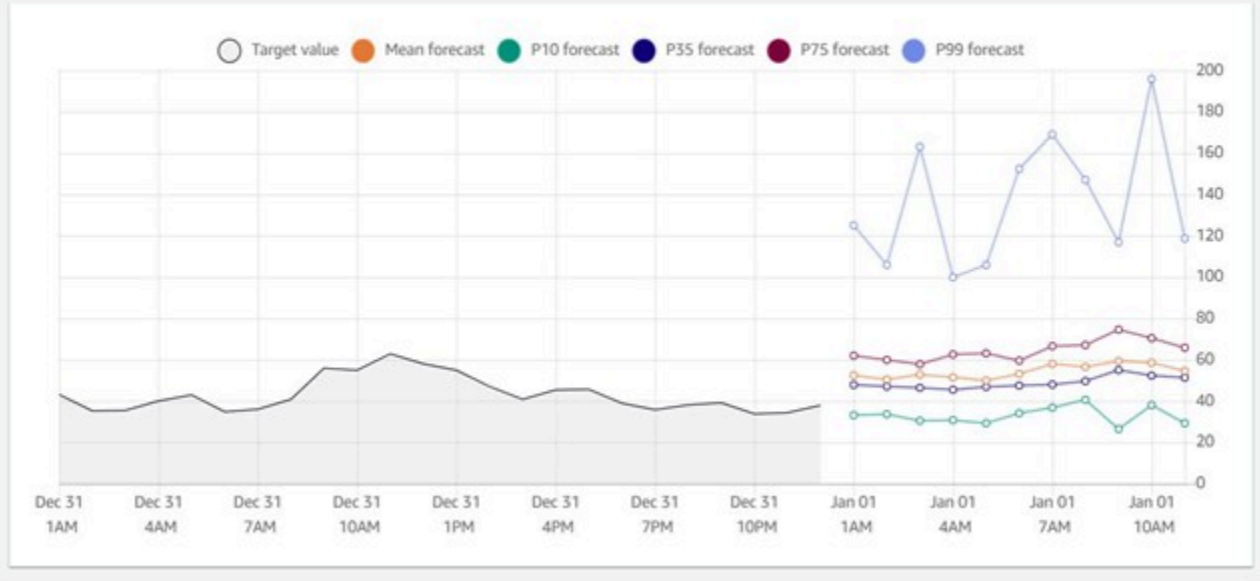
Di sisi lain, prakiraan P90 adalah prakiraan berlebih 90% dari waktu, dan prakiraan ini berguna ketika biaya peluang untuk tidak menjual item sangat tinggi, atau biaya modal yang diinvestasikan rendah. Untuk toko bahan makanan, prakiraan P90 mungkin digunakan untuk sesuatu seperti susu atau tisu toilet, di mana toko tidak pernah ingin kehabisan dan tidak keberatan selalu memiliki sisa item di rak.

Untuk prakiraan p50 (sering kali juga disebut prakiraan median), nilai sebenarnya diperkirakan lebih rendah dari nilai yang diprediksi 50% dari waktu, dan metrik wQL [0,5] dapat digunakan untuk menilai keakuratannya. Ketika kelebihan stok tidak terlalu memprihatinkan, dan ada jumlah permintaan yang moderat untuk item tertentu, prakiraan kuantil p50 bisa bermanfaat.

Visualisasi

Amazon Forecast memungkinkan pembuatan grafik prakiraan secara native di AWS Management Console. Selain itu, Anda dapat memanfaatkan tumpukan ilmu data Python lengkap (lihat [Contoh Amazon Forecast](#)). Amazon Forecast memungkinkan ekspor prakiraan sebagai file CSV melalui API `ExportForecastJob`, yang memungkinkan pengguna untuk memvisualisasikan prakiraan dalam alat analitik pilihan mereka.

Item_id: client_12



Visualisasi yang disediakan di konsol Amazon Forecast pada kuantil yang berbeda

Ringkasan alur kerja prakiraan dan API

Tabel berikut mencocokkan setiap langkah alur kerja prakiraan dengan masing-masing API Amazon Forecast.

Tabel 1: Langkah-langkah prakiraan dan API Amazon Forecast

Langkah	API	Fungsi API
Langkah 1: Pengumpulan dan agregat data Langkah 2: Persiapan data	Create_Dataset_Group , Create_Dataset , Create_Dataset_Import_Job	<ol style="list-style-type: none"> 1. Menetapkan domain tingkat tinggi (retail, metrik, dan sebagainya) untuk masalah. 2. Menentukan skema untuk set data yang berbeda (target, terkait, item-meta data). 3. Mengimpor data dari Amazon S3 ke Amazon Forecast.
Langkah 3: Pembuatan prediktor Langkah 4: Evaluasi prediktor	Create_Predictor	<ol style="list-style-type: none"> 1. Melakukan ETL. 2. Membagi data menjadi set pelatihan/pengujian dan melatih model. 3. Secara opsional, menggunakan Create_predictor_backtest_Export_job untuk mengeksport hasil backtest ke CSV untuk menghitung metrik tingkat item.
Langkah 5: Pembuatan dan penggunaan prakiraan untuk pengambilan keputusan	Create_Forecast	<ol style="list-style-type: none"> 1. Melatih/hosting model. 2. Membuat prediksi atas jangkauan prakiraan untuk kuantil tertentu yang

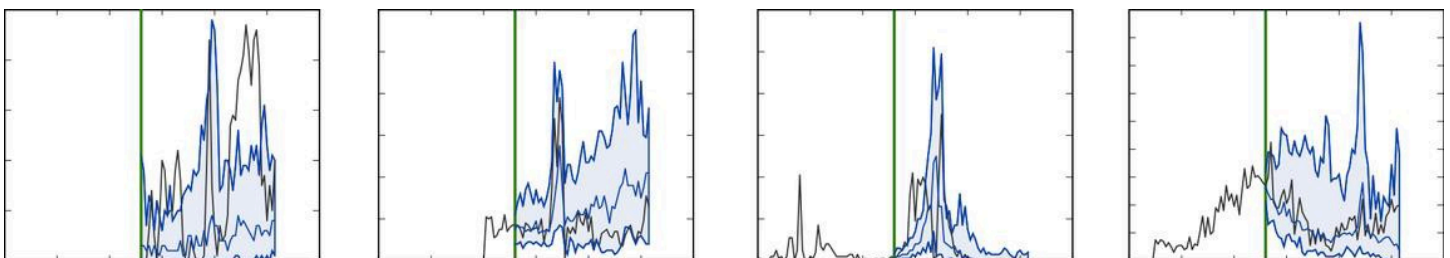
Langkah	API	Fungsi API
		diminati (misalnya, bilangan bulat antara 1 hingga 99 termasuk rata-rata).
	Query_Forecast . Create_Forecast_Export_Job	Memungkinkan Anda untuk mengonsumsi prakiraan yang dibuat oleh Create_Forecast

Menggunakan Amazon Forecast untuk skenario umum

Anda juga dapat melakukan analisis bagaimana-jika dengan membuat prakiraan yang berbeda berdasarkan perubahan variabel eksternal (misalnya, harga atau promosi). Misalnya, dalam contoh studi kasus e-commerce, Anda dapat membuat prakiraan yang berbeda berdasarkan promosi yang mungkin Anda rencanakan. Anda dapat memperkirakan permintaan untuk produk dengan diskon 10% kemudian dengan diskon 20% untuk memahami jumlah produk yang akan Anda perlukan untuk stok agar memenuhi permintaan. Hal ini dapat dicapai dengan menyiapkan grup set data unik dan memperbarui deret waktu terkait di masing-masing, berdasarkan skenario yang diminati.

Selain itu, Anda juga dapat membuat prakiraan untuk item tanpa riwayat sebelumnya (kadang-kadang disebut masalah cold start). Pendekatan ini membutuhkan pembuatan prediktor menggunakan DeepAR+ atau CNN-QR bersama dengan metadata (seperti set data metadata item) untuk membuat prakiraan item baru.

Gambar berikut menunjukkan contoh empat SKU berbeda yang muncul dalam masalah prakiraan operasional dunia nyata.



Contoh empat SKU berbeda yang muncul dalam masalah prakiraan operasional dunia nyata.

Pada gambar sebelumnya, data aktual yang diamati berada di sebelah kiri garis vertikal, dan prakiraan berwarna biru berada di sebelah kanan garis vertikal, dibandingkan dengan data aktual berwarna hitam. Perhatikan bahwa riwayat SKU masing-masing individu, di sebelah kiri garis vertikal, tidak menunjukkan evolusinya di sebelah kanan garis hijau.

Menerapkan Prakiraan ke dalam produksi

Setelah menyelesaikan alur kerja Amazon Forecast menyeluruh, sangat penting untuk mengidentifikasi perbedaan utama antara API `Create_Predictor` dan `Create_Forecast` dan kapan masing-masing harus digunakan.

Yang pertama digunakan terutama selama bukti konsep untuk mengevaluasi akurasi/metrik model, sedangkan yang terakhir digunakan untuk membuat prakiraan dalam lingkungan produksi.

Setelah dalam produksi, `Create_Predictor` tidak perlu dijalankan setiap kali prakiraan harus dibuat, tetapi hanya ketika model perlu dilatih ulang karena perubahan data atau sebagai bagian dari frekuensi yang telah ditetapkan sebelumnya (misalnya, setiap dua minggu atau setiap bulan). Karena set data diperbarui dengan data baru, hanya `Create_Forecast` perlu dijalankan untuk membuat prakiraan untuk jangkauan prakiraan baru.

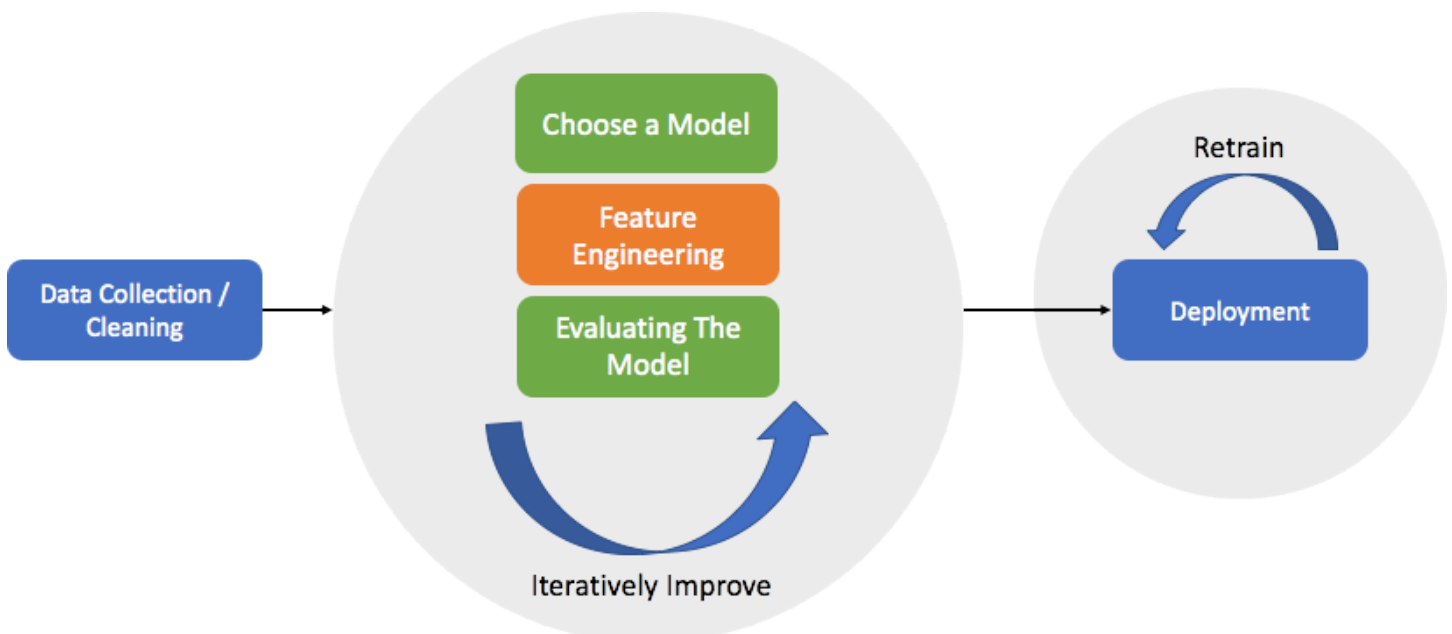
Dalam produksi, Anda juga perlu mengotomatiskan impor set data dan operasi prakiraan untuk menghasilkan prediksi baru secara bergulir. Saat ini, otomatisasi ini dapat dicapai dengan menyiapkan pekerjaan cron menggunakan kombinasi log [Amazon CloudWatch Events](#), [AWS Step Functions](#), dan fungsi [AWS Lambda](#). Pengaturan pekerjaan [cron](#) pada gilirannya mengotomatiskan panggilan API Amazon Forecast untuk impor/pelatihan ulang atau pembuatan prakiraan. Terakhir, penting untuk mengelola sumber daya Anda dan menghapusnya secara berkala sehingga Anda tidak melebihi [batas sistem](#) yang ditentukan oleh layanan. Lihat [posting blog](#) ini, termasuk [Amazon Redshift](#) untuk mempelajari lebih lanjut tentang menyiapkan pekerjaan terjadwal.

Kesimpulan

[Januschowski dan Kolassa \(2019\)](#) memberikan klasifikasi tentang masalah pemrakiraan yang selaras dengan keputusan yang perlu diambil bisnis, termasuk keputusan strategis, taktis, dan operasional. Setiap tingkat keputusan memiliki tugas prakiraan yang bersangkutan.

Masalah prakiraan operasional dan taktis memiliki ciri khas terdiri dari data dalam jumlah besar, dan biasanya memerlukan banyak otomatisasi. Metode prakiraan yang berbeda cocok untuk masalah ini. Metode prakiraan lokal biasanya berfungsi dengan baik untuk masalah prakiraan strategis, metode berbasis deep learning untuk masalah prakiraan operasional, dan untuk di antaranya, mungkin diperlukan sedikit eksperimen. Meskipun laporan ini membahas masalah prakiraan operasional, Amazon Forecast tidak bersikeras tentang model yang ditawarkannya dan mencakup model yang cocok untuk masalah prakiraan strategis serta operasional dan taktis.

Proses pemecahan masalah prakiraan operasional dapat dipecah menjadi langkah-langkah dasar mulai dari pengumpulan data dan persiapan hingga pembuatan model dan deployment. Secara umum, menganggap hal ini sebagai proses berulang dan bukannya proses linier merupakan tindakan yang paling bermanfaat. Misalnya, karena model dan kasus penggunaan lebih dipahami dengan baik, kembali ke tahap pengumpulan data mungkin masuk akal untuk dilakukan. Pengembangan model itu sendiri juga merupakan tindakan yang sangat berulang.



Proses pengembangan yang disederhanakan dalam membawa model prakiraan ke produksi.

Kontributor

Kontributor dokumen ini antara lain:

- Yuyang Wang, Ilmuwan Senior Machine Learning (Senior Machine Learning Scientist), Layanan Vertikal AI
- Danielle Robinson, Ilmuwan Terapan (Applied Scientist), Layanan Vertikal AI
- Tim Januschowski, Manajer (Manager), Ilmu Terapan ML
- Namita Das, Manajer Produk Senior (Senior Product Manager), Layanan Vertikal AI
- Christy Bergman, Spesialis AI/ML Senior (Senior AI/ML Specialist), Arsitek Solusi
- Kris Tonthat, Penulis Teknis (Technical Writer), Dokumentasi AI/ML

Sumber Bacaan Lebih Lanjut

Untuk informasi tambahan tentang prakiraan deret waktu dan metode deep learning, lihat:

- [Dokumentasi Amazon Forecast](#)
- [Blog ketersediaan umum Amazon Forecast](#)
- [Sekarang tersedia di Amazon SageMaker: Algoritme DeepAR agar lebih akurat](#)
- [Amazon SageMaker DeepAR sekarang mendukung nilai yang hilang, fitur deret waktu dan kategoris, serta frekuensi umum](#)
- [Amazon Forecast sekarang dapat menggunakan Convolutional Neural Networks \(CNN\) untuk melatih model prakiraan hingga 2X lebih cepat dengan akurasi hingga 30% lebih tinggi](#)
- [Amazon Forecast sekarang mendukung pengukuran akurasi untuk masing-masing item](#)
- [Mengukur akurasi model prakiraan untuk mengoptimalkan tujuan bisnis Anda dengan Amazon Forecast](#)
- [Amazon Forecast Weather Index — secara otomatis menyertakan cuaca lokal untuk meningkatkan akurasi model prakiraan Anda](#)
- [Makalah ilmiah tentang model prakiraan deret waktu](#)
- [Halaman GitHub sampel Amazon Forecast](#)
- [AWS Pusat Arsitektur](#)

Lampiran A: FAQ

T: Apa yang harus saya lakukan untuk memulai dengan Amazon Forecast?

1. Pertama, Anda akan membutuhkan Akun AWS.
2. Selanjutnya, buka layanan Forecast di [AWS Management Console](#), buat grup set data, dan impor file .csv ke set data deret waktu target (wajib). Data minimum yang diperlukan untuk memulai adalah data historis untuk jumlah yang ingin Anda prediksi, seperti listrik per tanda waktu per rumah tangga.
3. Terakhir, buat model dengan menjalankan [CreatePredictor](#) dan mengeluarkan hasil dengan menjalankan [CreateForecast](#). Lihat halaman dokumentasi [Memulai](#) untuk selengkapnya.

Lihat juga [panduan Pendahuluan dan Praktik Terbaik GitHub](#).

T: Apakah Amazon Forecast cocok untuk saya?

Tidak semua masalah machine learning merupakan masalah prakiraan. Pertanyaan pertama yang harus ditanyakan adalah, “Apakah masalah bisnis saya termasuk deret waktu dalam pernyataannya?” Misalnya, apakah Anda memerlukan nilai tertentu hanya pada waktu dan tanggal tertentu di masa mendatang? Prakiraan tidak cocok untuk masalah umum, statis (di mana tanggal/waktu tertentu tidak penting), seperti deteksi penipuan atau untuk merekomendasikan judul film kepada pengguna. Ada solusi yang jauh lebih cepat untuk masalah statis.

Selain memiliki data deret waktu, data itu sendiri harus “padat,” dan memiliki riwayat panjang. Hal ini dirangkum dalam tabel berikut:

Tabel 2 — Kriteria dan kelas algoritme Amazon Forecast

Kriteria	Kelas algoritme Amazon Forecast
Set data besar dengan hingga lima juta deret waktu dengan pola dasar serupa + efek musiman + data terkait. Setiap deret waktu harus memiliki riwayat panjang, idealnya lebih dari dua tahun jika mencoba untuk mendapatkan peristiwa tahunan, dan setiap deret waktu	DeepAR+, CNN-QR deep learning eksklusif Amazon Forecast

Kriteria	Kelas algoritme Amazon Forecast
yang lebih dari 300, idealnya setidaknya 1K titik data.	
Set data kecil dengan 1-100 deret waktu, di mana sebagian besar deret waktu memiliki lebih dari 300 titik data + efek musiman + data terkait.	Prophet
Set data kecil dengan 1-10 deret waktu, di mana sebagian besar deret waktu memiliki lebih dari 300 titik data + efek musiman.	ETS, ARIMA
Intermiten (jarang mengandung banyak 0) dengan 1-10 deret waktu, di mana sebagian besar deret waktu memiliki lebih dari 300 titik data.	NPTS eksklusif Amazon Forecast
Set data kecil (reguler atau jarang) dengan 1-10 deret waktu, di mana sebagian besar deret waktu memiliki kurang dari 300 titik data.	Data terlalu kecil untuk Amazon Forecast. Coba ETS di Excel atau model statistik tradisional ARIMA dan Prophet sebagai gantinya.

Praktik terbaik adalah berlatih dengan mode AutoML di Predictor Anda, pertama kalinya pada data Anda. AutoML secara otomatis akan beroperasi melalui semua algoritme, (algoritme DL dijalankan dengan HPO diaktifkan), untuk mempelajari algoritme mana yang berfungsi paling baik pada data Anda.

T: Apa pendapat saya tentang data yang hilang? Kapan membuat prakiraan yang masuk akal itu terlalu merepotkan?

Mungkin saja ada masalah dalam perekaman data atau tingkat agregasi data terlalu rendah atau terlalu tinggi. Aturan umumnya adalah, panjang Forecast tidak boleh lebih dari 1/3 data pelatihan.

Selain jumlah data yang hilang, pertimbangan lain adalah imputasi data yang hilang. Anda dapat mengonversi semua 0 menjadi null dan membiarkan Amazon Forecast melakukan tugas berat yakni mengimputasi nilai-nilai yang hilang secara otomatis. Amazon Forecast akan secara otomatis mendeteksi apakah nilai yang hilang terjadi karena pengenalan produk baru (cold-start) atau produk pada akhir masa pakai. Anda dapat menggunakan beberapa logika nilai yang hilang termasuk nilai,

median, minimum, maksimum, nol, rata-rata, dan nan (deret waktu target saja. Lihat [dokumentasi untuk null-filling syntax](#).

- “frontfill” - (hanya TTS) mengacu pada item baru atau cold-start dan bagaimana Anda ingin memperlakukan nulls sebelum item mulai memiliki riwayat
- “middlefill” - Mengacu pada nulls di tengah nilai deret waktu
- “backfill” - Mengacu pada item pada akhir masa pakai dan bagaimana Anda ingin memperlakukan nulls setelah item berhenti laku
- “futurefill” - (hanya RTS) mengacu pada nulls yang terjadi setelah akhir data pelatihan

T: Data historis input saya tidak memiliki nilai negatif, tetapi saya melihat nilai negatif dalam prakiraan permintaan? Mengapa ini terjadi? Apa yang bisa saya lakukan untuk menghindari hal ini?

Untuk semua model selain NPTS (dilatih pada data non-negatif) dan DeepAR (dengan fungsi kemungkinan negatif-binomial), tidak ada jaminan akan menghasilkan angka positif. Solusinya adalah mengubah ke salah satu model yang disebutkan di atas, atau memotong nilai prakiraan ke nilai non-negatif.

T: Mengapa metrik akurasi berbeda pada kuantil? Bukankah kesalahannya sama karena modelnya sama?

Lihat [Weighted Quantile Loss \(wQL\)](#) untuk penjelasan lebih lanjut tentang bagaimana pembobotan bergantung pada kuantil.

Bayangkan Anda memiliki semua prakiraan pada tiga kuantil yang berbeda: p10, p50, p90. Tiga prediksi itu sendiri adalah variabel acak. Akurasi dihitung secara terpisah antara aktual dan prakiraan pada setiap tingkat kuantil. Anda mungkin akan melihat tabel “wQL”, weighted quantile loss, seperti di bawah ini. Nilai-nilai wQL tidak memiliki hubungan yang menentukan antara satu sama lain. (Ingat loss berarti kesalahan, sehingga tidak diorder; tetapi prakiraan kuantil diorder). Jadi, misalnya, tidak ada alasan mengapa p90 wQL harus lebih besar dari p50 wQL.

Table 3 — Contoh prakiraan kuantil

	A	B	C
1	P10 wQL	P50 wQL	P90 wQL
2	0,18647	0,50879	0,30428

T: Bagaimana cara meningkatkan akurasi prakiraan?

Akurasi prakiraan bergantung pada apakah data yang tepat tersedia dalam kuantitas dan kualitas yang tepat. Jika akurasi tidak memuaskan, mungkin masuk akal untuk memahami seberapa dapat diprediksi masalahnya (atau seberapa acak/salah/stasioner datanya). Faktor lain yang perlu dipertimbangkan termasuk mengevaluasi model yang berbeda, pengaturan hiperparameter, dan menyertakan fitur tambahan dengan menggunakan set data metadata item dan deret waktu terkait. Untuk saran spesifik, [lihat dokumen Praktik Terbaik ini di GitHub](#).

T: Saya memiliki algoritme yang berfungsi sangat baik untuk kasus penggunaan saya, dan tidak ditawarkan di Amazon Forecast. Apa yang harus saya lakukan?

Tim Amazon Forecast akan dengan senang hati membantu Anda dalam kasus penggunaan ini. Hubungi tim layanan Amazon Forecast dengan mengirim email ke amazonforecast-poc@amazon.com.

Lampiran B: Referensi

[Januschowski, Tim dan Kolassa, Stephan. A classification of business forecasting problems. Foresight: The International Journal of Applied Forecasting. 2019](#)

[Salinas, David, Flunkert, Valentin, Gasthaus, Jan, dan Januschowski, Tim. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting. 2019](#)

[Gasthaus, Jan, Benidis, Konstantinos, Wang, Yuyang, Rangapuram, Syama Sundar, Salinas, David, Flunkert, Valentin, dan Januschowski, Tim. {Probabilistic Forecasting with Spline Quantile Function RNNs. The 22nd International Conference on Artificial Intelligence and Statistics. 2019](#)

[Januschowski, Tim, Gasthaus, Jan, Wang, Yuyang, Salinas, David, Flunkert, Valentin, Bohlke-Schneider, Michael, dan Callot, Laurent. Criteria for classifying forecasting methods. International Journal of Forecasting. 2019 \(Wajib sign-in\)](#)

[Januschowski, Tim, Gasthaus, Jan, Wang, Yuyang, Rangapuram, Syama, dan Callot, Laurent. Deep Learning for Forecasting. Foresight: The International Journal of Applied Forecasting. 2018](#)

[Januschowski, Tim, Gasthaus, Jan, Wang, Yuyang, Rangapuram, Syama Sundar, dan Callot, Laurent. Deep Learning for Forecasting: Current Trends and Challenges. Foresight: The International Journal of Applied Forecasting. 2018](#)

[Bose, Joos-Hendrik, Flunkert, Valentin, Gasthaus, Jan, Januschowski, Tim, Lange, Dustin, Salinas, David, Schelter, Sebastian, Seeger, Matthias, dan Wang, Yuyang. Probabilistic demand forecasting at scale. Proceedings of the VLDB Endowment. 2017](#)

Riwayat dokumen

Untuk mendapatkan notifikasi tentang pembaruan laporan resmi ini, sebaiknya berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Laporan resmi diperbarui	Pembaruan.	1 September 2021
Publikasi awal	Laporan resmi pertama kali diterbitkan.	4 Februari 2020

Note

Untuk berlangganan pembaruan RSS, Anda harus mengaktifkan plug-in RSS untuk browser yang Anda gunakan.

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya disediakan sebagai informasi, (b) berisi AWS penawaran produk dan praktik saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak menjadi komitmen atau jaminan apa pun dari AWS dan afiliasi, pemasok, atau pemberi lisensinya. Produk atau layanan AWS diberikan “apa adanya” tanpa jaminan, pernyataan, atau syarat apa pun, baik secara tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

© 2021 Amazon Web Services, Inc. atau afiliasinya. Semua hak dilindungi undang-undang.

AWS glosarium

Untuk terminologi AWS terbaru, lihat [glosarium AWS](#) di Referensi Umum AWS.